Parallel Ports, Power Supply, and the Clock Oscillator

Chapter 3

Dr. Iyad Jafar
Outline

• Why Do We Need Parallel Ports?
• Hardware Realization of Parallel Ports
• Interfacing to Parallel Ports
• The PIC 16F84A Parallel Ports
• The Power Supply
• The Clock Oscillator
Why Do We Need Parallel Ports?

- Almost any microcontroller needs to transfer digital data from/to external devices and for different purposes
  - Direct user interface – switches, LEDs, keypads, displays
  - Input measurement - from sensors, possibly through ADC
  - Output control information – control motors and actuators
  - Bulk data transfer – to other systems/subsystems
- Transfer could be serial or parallel! Analog or digital!
The PIC 16F84 Parallel Ports

Port A, bit 2  RA2
Port A, bit 3  RA3
*Port A, bit 4 RA4/T0CKI
  Reset          MCLR
  Ground         VSS
**Port B, bit 0 RB0/INT
  Port B, bit 1  RB1
  Port B, bit 2  RB2
  Port B, bit 3  RB3
  9
  10
  RB4

*also counter/timer clock input
**also external interrupt input

Oscillator connections
Supply voltage
The PIC 16F84 Parallel Ports

PORT A

- 5-bit general-purpose bidirectional digital port

Related registers

- Data from/to this port is stored in PORTA register (0x05)
- Pins can be configured for input or output by setting or clearing corresponding bits in the TRISA register (0x85)

- Pin RA4 is multiplexed and can be used as the clock for the TIMER0 module
The PIC 16F84 Parallel Ports

PORT B

- **8-bit general-purpose** bidirectional digital port
- Related registers
  - Data from/to this port is stored in `PORTB` register (0x06)
  - Pins can be configured for input or output by setting or clearing, corresponding bits in the `TRISB` register (0x86), respectively
- Other features
  - Pin `RB0` is multiplexed with the external interrupt INT and has Schmitt trigger interface
  - Pins `RB4 – RB7` have a useful ‘interrupt on change’ facility
Example 1 – configuring port B such that pins 0 to 2 are inputs, pins 3 to 4 outputs, and pins 5 to 7 are inputs

bsf STATUS, RP0 ; select bank1
movlw 0xE7
movwf TRISB ; PORTB<7:5> input,
            ; PORTB<4:3> output
            ; PORTB<2:0> input
The PIC 16F84 Parallel Ports

- **Example 2** – configuring PORTB as output and output value 0xAA

  - `bsf STATUS, RP0 ; select bank1`
  - `clrf TRISB ; PORTB is output`
  - `movlw 0xAA`
  - `bcf STATUS, RP0 ; select bank0`
  - `movwf PORTB ; output data`

- **Example 3** – configuring PORTA as input, read it and store the value in 0x0D

  - `bsf STATUS, RP0 ; select bank1`
  - `movlw 0xFF`
  - `movwf TRISA ; PORTA is input`
  - `bcf STATUS, RP0 ; select bank0`
  - `movf PORTA, W ; read data`
  - `movwf 0x0D ; save data`
Interfacing to Parallel Ports

Switches

Interfacing to SPDT switch. A current limiting resistor might be needed.

Interfacing to SPST switch. To reduce wasted current, the pull-up resistor $R$ should be high (10-100KOhms).

Interfacing to SPST switch using a pull-down resistor.
Interfacing to Parallel Ports

Light Emitting Diodes (LEDs)

- LEDs can be driven from a logic output as long as the current requirements are met. Interfacing of LEDs depending on the logic type and their capability to source and sink current.

\[
V_{OH} \text{ Logic gate output high voltage}
\]

\[
V_{OL} \text{ Logic gate output low voltage}
\]

For current source:
\[
V_{OH} = R I_D + V_D
\]
\[
R = \frac{V_{OH} - V_D}{I_D}
\]

For current sink:
\[
V_S = V_{OL} + R I_D + V_D
\]
\[
R = \frac{V_S - V_D - V_{OL}}{I_D}
\]
Interfacing to Parallel Ports

Light Emitting Diodes (LEDs)

- A special type of diodes made of semiconductor material that can emit light when forward biased

![Graph showing LED characteristics](image)

- **Type number:** L-441D
  - Wavelength = 627 nm
  - 15mcd typ. @ 10 mA

- **Type number:** L-44GD
  - Wavelength = 565 nm
  - 12mcd typ. @ 10 mA
Interfacing to Parallel Ports

7-Segment Display

<table>
<thead>
<tr>
<th>Digit Shown</th>
<th>Illuminated Segment (1 = illumination)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>
Interfacing to Parallel Ports

Port Electrical Characteristics

- Logic gates are designed to interface easily with each other, especially when connecting gates from the same family.
- The concern arises when connecting logic gates to non-logic devices such as switches and LEDs.

Generalized model

CMOS model
Interfacing to Parallel Ports

Light Emitting Diodes (LED)

Computation of limiting resistors when internal resistance of the port pin is considered
The PIC 16F84 Parallel Ports

Port Output Characteristics

\[ V_{OH} \text{ vs. } I_{OH} \ (VDD = 3V, -40 \text{ to } 125^{\circ}C) \]

\[ R_{OH} = 130 \ \Omega \]
The PIC 16F84 Parallel Ports

Port Output Characteristics

\( V_{OL} \) vs. \( I_{OL} \) (\( VDD = 3V, -40 \text{ to } 125^\circ C \))

\[ R_{OL} = 36 \, \Omega \]
Example 3.1

- **Example** – Write a program that continuously reads an input value from 4 switches connected to PORTA (RA3-RA0) and display the value on 4 LEDs connected to PORTB (RB7-RB4). Make sure to draw the circuit and configure the ports properly.

- **Requirements:**
  
  1) **Connect four switches to RA3-RA0. Configure these pins as input.**
  
  2) **Connect four LEDs to RB7-RB4. Configure these pins as outputs.**
Example 3.1

```
#include "P16F84A.INC"

TEMP EQU 0X20
ORG 0X0000

; ------------------- MAIN PROGRAM -------------------

MAIN
  BSF STATUS, RP0 ; SELECT BANK 1
  MOVLW B'00001111'
  MOVWF TRISA ; CONFIGURE RA3-RA0 AS INPUT
  MOVLW B'00001111'
  MOVWF TRISB ; CONFIGURE RB7-RB4 AS OUTPUT
  BCF STATUS, RP0

REPEAT
  MOVF PORTA, W ; READ FROM PORT A
  ANDLW 0X0F ; MASK THE LOWER 4 BITS IN PORTA
  MOVWF TEMP
  SWAPF TEMP, W ; MOVE BITS TO RB7-RB4
  MOVWF PORTB

GOTO REPEAT
END
```
Example 3.2

- **Example** – Modify the program and the circuit in Example 3.1 such that the switches are read and displayed when an external interrupt occurs (falling edge) only.

- **Requirements:**
  1) **Connect four switches to RA3-RA0. Configure these pins as input.**
  2) **Connect four LEDs to RB7-RB4. Configure these pins as outputs.**
  3) **Connect a switch to RB0 and configure it as input.**
Example 3.2

```assembly
.include "P16F84A.INC"

TEMP EQU 0X20
ORG 0X0000
GOTO MAIN
ORG 0X0004
GOTO ISR

; ----------------------- MAIN PROGRAM -----------------------

MAIN
    BSF STATUS, RP0 ; SELECT BANK 1
    MOVLW B'00001111'
    MOVWF TRISA ; CONFIGURE RA3-RA0 AS INPUT
    MOVLW B'00000001'
    MOVWF TRISB ; CONFIGURE RB0 AS INPUT
    BCF OPTION_REG, INTEDG ; INTERRUPT ON FALLING EDGE
    BCF STATUS, RP0
    BSF INTCON, INTE ; ENABLE INTERRUPT
    BSF INTCON, GIE
    GOTO WAIT ; WAIT FOR INTERRUPT

WAIT
    GOTO WAIT ; WAIT FOR INTERRUPT

ISR
    MOVF PORTA, W ; READ FROM PORT A
    ANDLW 0X0F ; MASK THE LOWER 4 BITS IN PORTA
    MOVWF TEMP
    SWAPF TEMP, W ; MOVE BITS TO RB7-RB4
    MOVWF PORTB
    BCF INTCON, INTE
    RETFIE
END
```
Example 3.3

- **Example** – Write a program to control the flashing of a LED that is connected to RB1 using a pushbutton that is connected to RB0. The LED starts flashing upon the arrival of the first rising edge on RB0. Afterwards, successive edges toggle the state of flashing (On, off, on, ...). When the LED is flashing, this implies that it is 0.5 second ON and 0.5 second OFF. Assume 4MHz clock.

- **Requirements:**
  1) *Configure RB0 as input and RB1 as output*
  2) *Enable external interrupt (INTE) and global interrupts (GIE)*
  3) *Write a 0.5 second delay routine*
  4) *Keep track of the current status of flashing (on/off)*
Example 3.3

```
#include "P16F84A.INC"

FLASH EQU 0X20 ; STORE THE STATE OF FLASHING
COUNT1 EQU 0X21 ; COUNTER FOR DELAY LOOP
COUNT2 EQU 0X22 ; COUNTER FOR DELAY LOOP

ORG 0X0000
GOTO START

ORG 0X0004
GOTO ISR

; ----------------------------------
; MAIN PROGRAM
; ----------------------------------

START
CLRF FLASH ; CLEAR FLASHING STATUS
BSF STATUS,RP0 ; SELECT BANK 1
MOVLW B'00000001' ; CONFIGURE RB0 AS INPUT AND RB1 AS OUTPUT
MOVWF TRISB
BSF OPTION_REG, INTEDG ; SELECT RISING EDGE FOR EXTERNAL INTERRUPT
BSF INTCON , INTE ; ENABLE EXTERNAL INTERRUPT
BSF INTCON , GIE ; ENABLE GLOBAL INTERRUPT
BCF STATUS,RP0 ; SELECT BANK 0
CLRF PORTB ; CLEAR PORTB; TURN OFF LED

WAIT
BTFSS FLASH , 0 ; IF BIT 0 OF FLASH IS CLEAR THEN NO FLASHING
GOTO WAIT ; WAIT UNTIL BIT 0 IS SET
MOVLW B'00000010'
XORWF PORTB , 1 ; COMPLEMENT RB1 TO FLASH
CALL DEL_p5sec
GOTO WAIT
```
Example 3.3

--------------- INTERRUPT SERVICE ROUTINE ---------------
ISR
  MOVLW 0x01  
XORWF FLASH, F ; COMPLEMENT THE STATUS  
BCF INTCON, INTF ; CLEAR THE INTF FLAG  
RETFIE

; -------------- DELAY ROUTINE -------------------------
DEL_p5sec
  MOVLW D'0'  
  MOVWF COUNT1  
  MOVVLW D'244'  
  MOVWF COUNT2  
LOOP
  NOP  
  NOP  
  NOP  
  NOP  
  NOP  
  NOP  
  DECFSZ COUNT1, F  
  GOTO LOOP  
  DECFSZ COUNT2, F  
  GOTO LOOP ; delay 0.500207 seconds
RETURN

END
Hardware Realization of Parallel Ports

Output Parallel Port

- **Read/Write**
- **Port Select**
- **Two lines of data bus**
- **D**
- **Q**
- **External pin**

**High whenever port address is selected**

**Flip-flop latches data bus value onto external pin, when memory location is selected, AND Write is active**
Hardware Realization of Parallel Ports

Input Parallel Port

Buffer transfers logic value on external pin onto data bus line, when memory location is selected, AND Read is active.
Hardware Realization of Parallel Ports

Bidirectional Parallel Port

![Diagram of Bidirectional Parallel Port](image_url)
Hardware Realization of Parallel Ports

PORT B
PINS RB3:RB0

Configurable pull-up resistors using RBPU bit in the OPTION register

Latches input data whenever the port is read

Multiplexed input

- **TRISB = ‘1’** enables weak pull-up (if RBPU = ‘0’ in the OPTION_REG register).
- **I/O pins have diode protection to VDD and Vss.**
Hardware Realization of Parallel Ports

PORT B
PINS RB7:RB4

- **Clearing the RBIF bit?**
- **Comparing previous and present port input values**

**Note:**
1. TRISB = '1' enables weak pull-up (if RBPU = '0' in the OPTION_REG register).
2. I/O pins have diode protection to VDD and VSS.
Hardware Realization of Parallel Ports

PORT A

Note: I/O pins have protection diodes to VDD and VSS.
Hardware Realization of Parallel Ports

Electrical Characteristics

- **Schmitt Trigger Input**
  - A special type of gate with two thresholds
  - Remove fluctuations and corruptions in the input signal
Hardware Realization of Parallel Ports

Electrical Characteristics

- **Open Drain Output**
  - Flexible style of output that can be adapted as a standard logic output or a direct drive for small loads

![Open Drain Output](image1)
![Open Drain Output Driving A Small Load](image2)
Hardware Realization of Parallel Ports

Electrical Characteristics

- Open Drain Output
  - Can be used as a wired-OR
The Oscillator

- The choice of clock determines the operating characteristics for the microcontroller
  - *Faster clock gives* faster execution, *but more power consumption*

- Accurate and stable operation of the microcontroller requires *accurate and stable clock*
The Oscillator
Oscillator types

Resistor–capacitor (RC).
• low cost
• not precise

Crystal or ceramic
• expensive
• stable and precise
• mechanically fragile
The PIC 16F84A Oscillator

- The 16F84A can be configured to operate in four different oscillator modes using the F0SC1 and F0SC0 in the configuration word.

<table>
<thead>
<tr>
<th>F0SC1</th>
<th>F0SC0</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>LP oscillator – <em>intended for low frequency</em> (≤ 200 KHz) <em>crystal application to reduce power consumption</em></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>XT oscillator – <em>standard crystal configuration</em> (1-4 MHz)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>HS oscillator – <em>high speed</em> (≥ 4 MHz)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>RC oscillator - <em>requires external resistor an capacitor</em></td>
</tr>
</tbody>
</table>
The PIC 16F84A Oscillator

- The 16F84A has two oscillator pins; OSC1 and OSC2.

**XT configuration**

**RC configuration**

**External Clock**
The PIC 16F84A Oscillator

- RC oscillator frequency dependence on power supply
## The Power Supply

<table>
<thead>
<tr>
<th>Param No.</th>
<th>Symbol</th>
<th>Characteristic</th>
<th>Min</th>
<th>Typ†</th>
<th>Max</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>D001</td>
<td>VDD</td>
<td>Supply Voltage</td>
<td>16LF84A</td>
<td>2.0</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
</tr>
<tr>
<td>D001A</td>
<td></td>
<td></td>
<td>16F84A</td>
<td>4.0</td>
<td>—</td>
<td>5.5</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RAM Data Retention Voltage (Note 1)</td>
<td>1.5</td>
<td>—</td>
<td>—</td>
<td>V</td>
<td>Device in SLEEP mode</td>
</tr>
<tr>
<td>D003</td>
<td>VPOR</td>
<td>VDD Start Voltage to ensure internal Power-on Reset signal</td>
<td>—</td>
<td>Vss</td>
<td>—</td>
<td>V</td>
<td>See section on Power-on Reset for details</td>
</tr>
<tr>
<td>D004</td>
<td>SVDD</td>
<td>VDD Rise Rate to ensure internal Power-on Reset signal</td>
<td>0.05</td>
<td>—</td>
<td>—</td>
<td>V/ms</td>
<td></td>
</tr>
<tr>
<td>D010</td>
<td>IDD</td>
<td>Supply Current (Note 2)</td>
<td>16LF84A</td>
<td>—</td>
<td>1</td>
<td>4</td>
<td>mA</td>
</tr>
<tr>
<td>D010A</td>
<td></td>
<td></td>
<td>16F84A</td>
<td>—</td>
<td>1.8</td>
<td>4.5</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>3</td>
<td>10</td>
<td>20</td>
<td>mA</td>
</tr>
<tr>
<td>D013</td>
<td></td>
<td></td>
<td>—</td>
<td>3</td>
<td>10</td>
<td>20</td>
<td>mA</td>
</tr>
<tr>
<td>D014</td>
<td></td>
<td></td>
<td>16LF84A</td>
<td>—</td>
<td>15</td>
<td>45</td>
<td>µA</td>
</tr>
</tbody>
</table>
The Power Supply

100 nF decoupling capacitor
Summary

- Parallel ports allow the exchange of data between the outside world and the CPU
- It is essential to understand the electrical characteristics and internal circuitry of ports
- All microcontrollers need a clock. The clock speed determine the power consumption
- Active elements of the oscillator are usually built inside the microcontroller and the designer selects the type and configure it
- It is a must to understand the power requirements of the microcontroller